



# Making packages



# 0. Inhalt:

1. Installing packages
2. Autoconf and automake
3. What you need
4. What you do
5. Additional Source

# 1. Installing


- GUI: ,aptitude‘
  - Additional info available
- Cmd: ,apt-get install NAME‘
  - Fast but user need to know name of tool to install
- Use repositories to install non-stable versions

## 2. Story of automake/-conf

- „Goldstandart“ for Unix-package distribution these days
- Automake: in the 70ies developed by Bell to update their non-commercial UNIX systems
- Autoconf: to address problems with version compatibility
  - Splitting up configuring and compiling of software in two steps to increase portability

### 3. What you need

- The code you want to distribute
- **,configure.in'** needed by autoconf to generate configure
- **,Makefile.am'** needed by automake to generate Makefile.in, which is finally used by make to manage the installation



```
(1)                                     [autoconf]
configure.in ----- > configure

(2)                                     [automake]
configure.in + Makefile.am ----- > Makefile.in

(3)                                     [configure]
Makefile.in ----- > Makefile
```

A simplified overview over the process of generating a Makefile.  
Update: Use `Configure.ac` and `,aclocal'` instead.

## 4. How to do it

- Makefile.am consists at least of:

```
bin_PROGRAMS = hello  
hello_SOURCES = hello.c
```

## ■ Configure.ac:

AC\_INIT ([name], [version], [info])

# AM\_INIT\_AUTOMAKE (Name, Version) (*old  
version not recommended*)

AM\_INIT\_AUTOMAKE

AC\_CONFIG\_FILES([Makefile])

AC\_OUTPUT



# Chain of tools:

- Aclocal
- Autoconf
- Automake - -add-missing
- ./configure *(lot of options to individualize)*
- => make && make install *(not yet a package!)*

## 5. Source

- Additional, german, easy to read:

<http://www.hs-augsburg.de/~hhoegl/oss/hausarbeiten-04/oss-autotools.pdf>